НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено
Завідувач кафедри
_____ О.В. Коваль
(підпис) (ініціали, прізвище)
“___”_____2019 р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

з напряму підготовки
6.050101 “Комп’ютерні науки’’

на тему: Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення найбільш потенційно прогресивних гравців/команд та їх рекомендації для відповідних команд

Виконав: студент 4 курсу, групи ТР-51

Донцов Михайло Дмитрович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник доцент, к.т.н. Коваль Олександр Васильович _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрям підготовки 6.050101 “Комп’ютерні науки’’

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О.В. Коваль
(підпис)
”___”_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту
Донцову Михайлу Дмитровичу
(прізвище, ім’я, по батькові)

1. Тема роботи “ Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення найбільш потенційно прогресивних гравців/команд та їх рекомендації для відповідних команд ”
керівник роботи доцент, к.т.н. Коваль Олександр Васильович
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”___”_____ 201__р. № _____

2. Строк подання студентом роботи ___ _____ 201__ р.
3. Вихідні дані до роботи персональний комп’ютер під керуванням операційної системи Microsoft Windows, мова програмування Python, середа розробки Sypder.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні рішення та можливі засоби реалізації взаємодії, обґрунтувати обрані програмні застосунки та шляхи розробки програмних додатків, розробити програмне забезпечення, розробити користувацький інтерфейс, зробити висновки за результатами роботи
5. Перелік ілюстраційного матеріалу (з точним зазначенням обов’язкових креслень) 1. Мета та завдання роботи. 2. Актуальність. 3. Алгоритм K-means. 4. Блок-схема алгоритму K-means. 5. Засоби програмної реалізації. 6. Архітектура. 7.Іфнтерфейс програмного продукту. 8.Висновки.

Дата видачі завдання ”___”_____ 201__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі	10.10.2018	
2.	Розробка архітектури та загальної структури системи	14.10.2018-23.12.2018	
3.	Розробка структур окремих підсистем	2.02.2019-3.03.2019	
4.	Підготовка матеріалів	4.03.2019-14.05.2019	
5.	Програмна реалізація системи	18.04.2019-14.05.2019	
6.	Захист програмного продукту	23.05.2019	
7.	Оформлення пояснювальної записки	27.05.2019-10.06.2019	
8.	Передзахист	27.05.2019	
9.	Захист	20.06.2019	

Студент _____ Донцов М. Д.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Коваль О. В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Метою роботи є створення системи для автоматизації процесу аналізу даних та розрахунків щодо визначення потенційно прогресивних гравців за вхідними параметрами. Для створення інтерфейсів використано засоби візуального програмування Qt Designer.

Було проведено аналіз алгоритмів щодо обробки інформації та визначено, який більш підходить для реалізації даної теми.

Результатом роботи стало створення програмного продукту, який розподіляє гравців на класи і визначає найбільш потенційних гравців.

Записка містить 56 сторінок, 17 рисунків, 3 формули, 3 додатки і 18 посилань.

Ключові слова: визначення потенційно прогресивних гравців, визначення найбільш прогресивних гравців, алгоритм K-Means, алгоритм k-середніх.

ABSTRACT

The purpose of the work is to create a system for automating the process of data analysis and calculations for identifying potentially progressive players according to the input parameters. Qt Designer visual programming tools are used to create interfaces.

An analysis of the information processing algorithms was performed and it was determined which is more suitable for the implementation of this topic.

The result of the work was the creation of a software product that distributes players to classes and identifies the most potential players.

The note contains 56 pages, 17 figures, 3 formulas, 3 attachments and 18 references.

Key words: definition of potentially progressive players, determination of the most progressive players, K-Means algorithm, k-medium algorithm.

ЗМІСТ

ВСТУП

На сьогоднішній день в кожній галузі людської діяльності використовуються різні програмні комплекси та системи для досягнення певних результатів відносно поставлених задач. У спорті вони використовуються для аналізу результатів спортсменів, а також для прогнозування їх подальших успіхів.

Останні роки розвивається сфера кіберспорту, і все більш людей захоплюється цим та спостерігає за командами у різних змаганнях. З'являється все більш команд та гравців, які хочуть спробувати та дізнатися на що вони здатні. Та за їх результатами, які можна проаналізувати для виявлення найбільш прогресивних команд або гравців.

Існують багато різних алгоритмів для аналізу великої кількості даних. Вже створені різноманітні бібліотеки та фреймворки, які надають змогу використовувати технології для власних проектів.

Виявлення потенційно прогресивних гравців є актуальною задачею на сьогоднішній день та її можна автоматизувати за допомогою різних існуючих технологій.

Тому, було запропоновано дослідити можливість застосування алгоритмів машинного навчання для виявлення найбільш прогресивних гравців та для подальшої їх рекомендації для різних команд.

1 ЗАДАЧА РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АНАЛІЗУ ІНФОРМАЦІЇ ЩОДО ВИЗНАЧЕННЯ ПОТЕНЦІЙНО ПРОГРЕСИВНИХ ГРАВЦІВ

У сучасному світі в останні десятиріччя широкого поширення набули проблеми систематизації інформації та визначення найбільш важливої. Існують різні сфери людської діяльності, які можна автоматизувати для зменшення затрат часу на вивчення та аналіз даних. Але в наш час все більше й більше набувають розповсюдження алгоритми машинного навчання, які імітують та моделюють діяльність людського мозку. Це надало нові величезні можливості для застосування комп’ютерів в аналізі та обробки великої кількості інформації для знаходження найбільш потенційно ефективної інформації.

Метою роботи є створення системи для автоматизації процесу аналізу даних та розрахунків щодо визначення потенційно прогресивних гравців за вхідними параметрами.

Робота розробленої системи демонструється для отримання кінцевого файлу в якому міститься інформація про гравців розподілених на декілька класів.

Розроблена система повинна забезпечувати наступні можливості:

— аналіз та обробка вхідних даних;

— отримання файлу.

Оскільки розроблена система може бути підсистемою комплексу аналізу даних або результатів гравців та команд модуль повинен мати:

— один вхід і один вихід. На вході програмний модуль отримує визначений набір початкових даних, виконує їх обробку і повертає один набір вихідних даних;

— функціональну завершеність. Модуль виконує набір визначених операцій для реалізації визначених операцій для реалізації кожної окремої функції, достатніх для завершення початої обробки даних;

— логічну незалежність. Результат роботи даного фрагменту програми не залежить від роботи інших модулів;

— слабкі інформаційні зв’язки з іншими програмними модулями. Обмін інформацією між окремими модулями повинен бути мінімальним;

2 АКТУАЛЬНІСТЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

На сьогоднішній день кіберспорт набуває популярності серед молодого покоління людей та з кожним роком займає все більшу і більшу частину ринку.

Враховуючи тенденції розвитку кіберспорту у майбутньому може стати одним із головних спортивних дисциплін.

2.1 Аналіз перегляду онлайн трансляцій

Якщо подивитися на графіки які представлені нижче, то можна зрозуміти що на даний момент люди зацікавлені переглядом змагань.

З нижче наведених даних можна зрозуміти, що досить стабільна кількість глядачів переглядає змагання та ця цифра з кожним роком зростає.

На рисунках 2.1, 2.2, 2.3, 2.4 зображено графіки перегляду змагання.

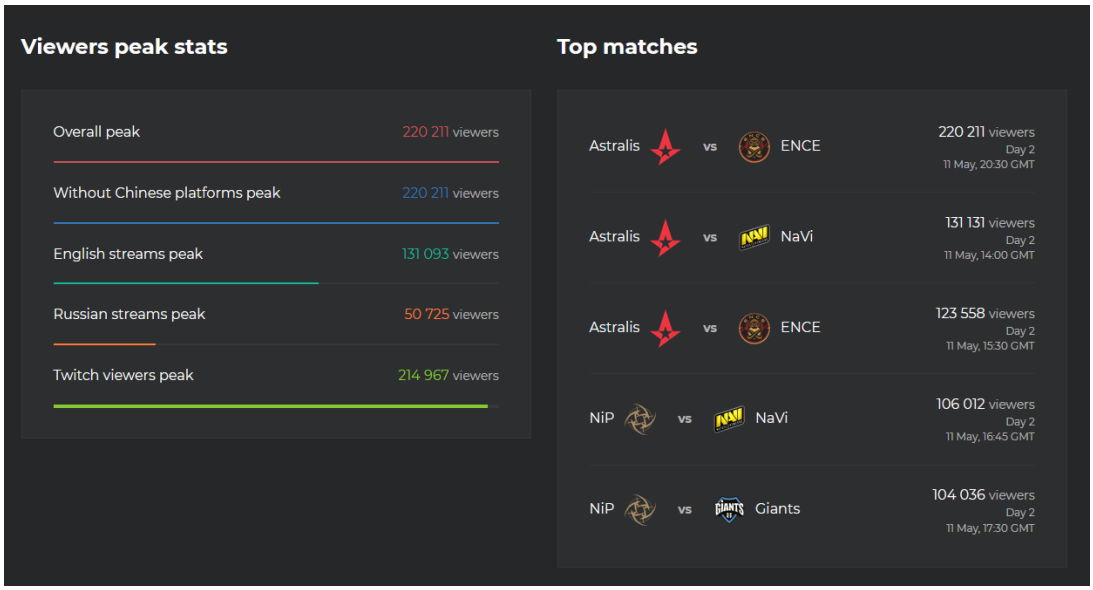


Рисунок 2.1 — Перший графік перегляду онлайн трансляцій

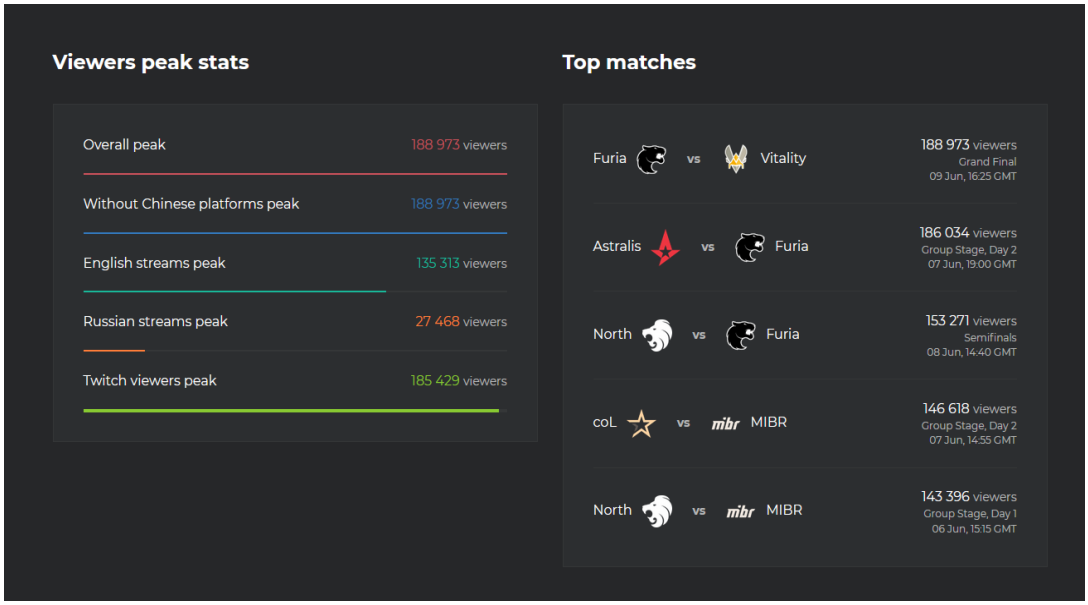


Рисунок 2.2 — Другий графік перегляду онлайн трансляцій

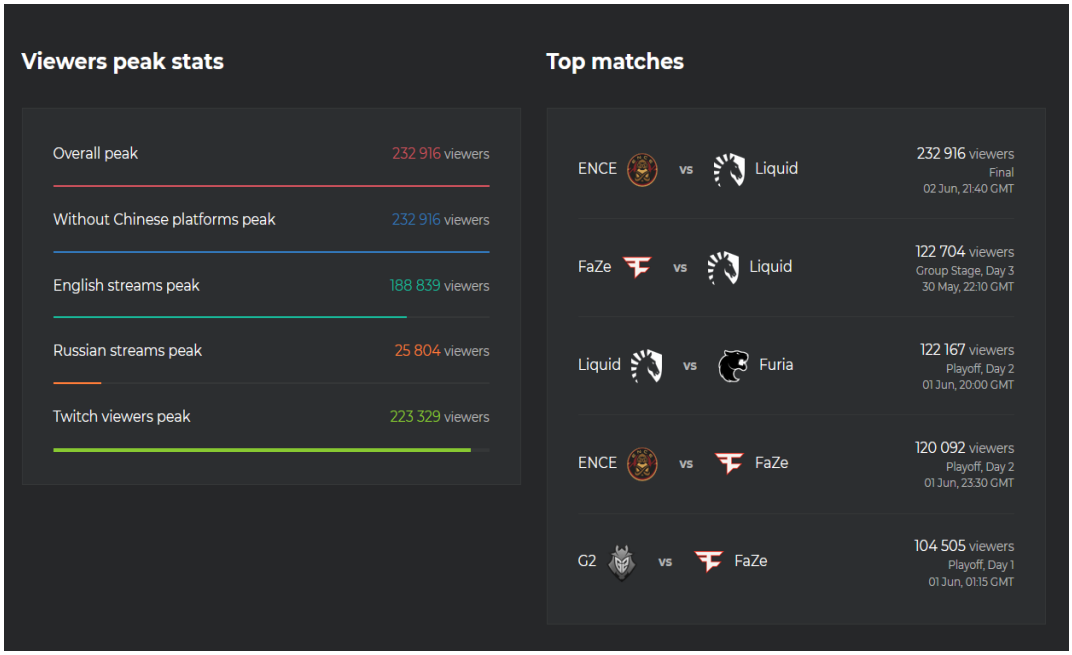


Рисунок 2.3 — Третій графік перегляду онлайн трансляцій

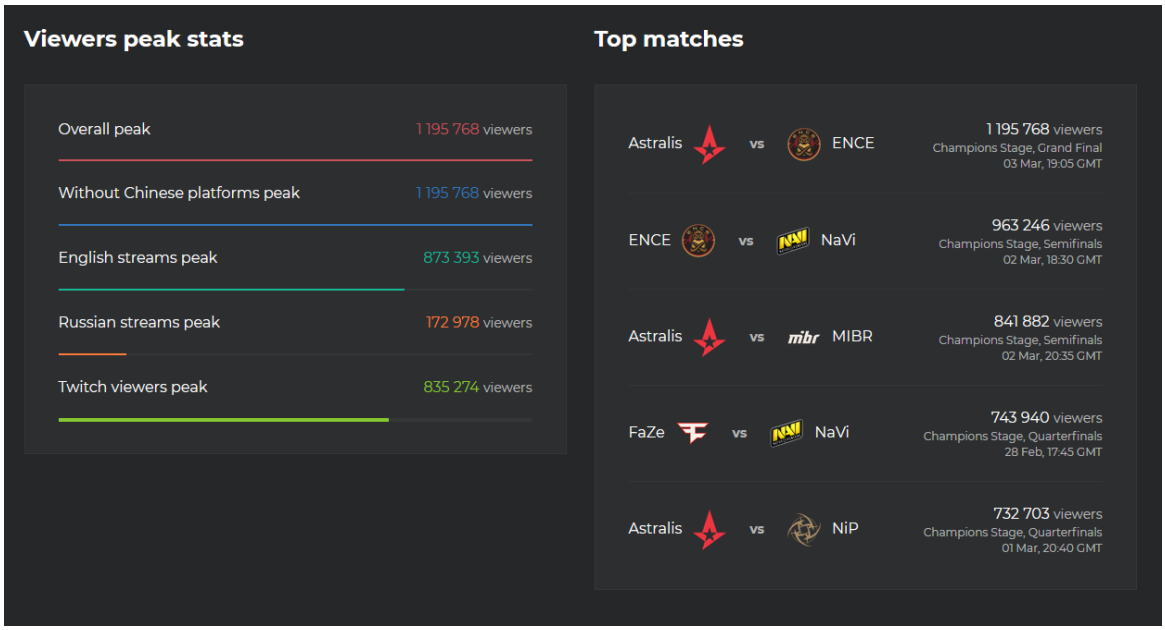


Рисунок 2.4 — Четрвертий графік перегляду онлайн трансляцій

На рисунках 2.5 та 2.6 зображено популярність вводу у пошуковий рядок сервіса Google слів «Counter strike Global offensive» та «кіберспорт».

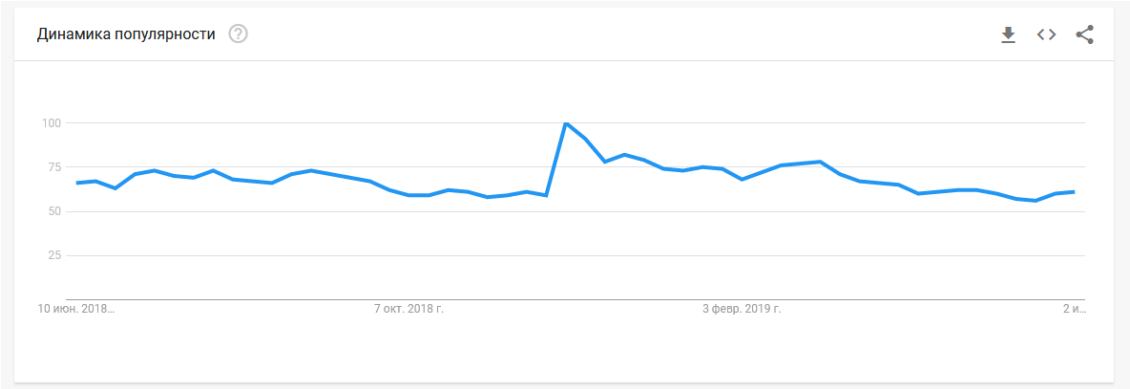


Рисунок 2.5 — Динаміка популярності слів кіберспортивної гри

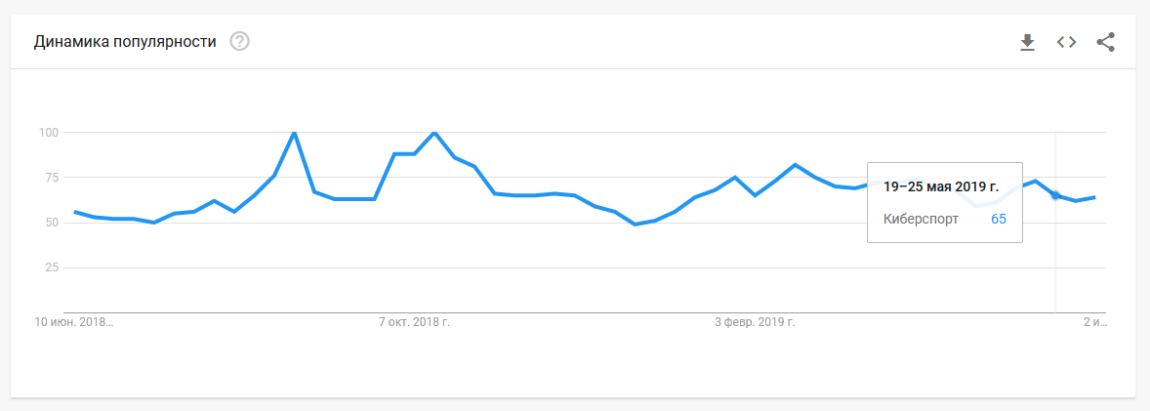


Рисунок 2.6 — Динаміка популярності слова кіберспор

Можна зробити висновок, що люди досить стабільно шукають ключові слова та популярність не спадає.

3 АНАЛІЗ АЛГОРИТМІВ АНАЛІЗУ ІНФОРМАЦІЇ ЩОДО ВИЗНАЧЕННЯ ПОТЕНЦІЙНО ПРОГРЕСИВНИХ ГРАВЦІВ

З кожним роком набувають все більшого поширення різноманітні алгоритми машинного навчання. Вони надають змогу для створення різноманітного програмного забезпечення, яке вирішує широкий спектр задач. Можна виділити задачу про аналіз даних.

З метою автоматизації задачі визначення потенційно прогресивних гравців було запропоновано розробити програмну систему для її вирішення. Також розглянемо існуючи системи і сфери застосування.

3.1 Машинне навчання

Машинне навчання є галуззю штучного інтелекту в комп'ютерних технологіях, що використовує статистичні методи, щоб дозволити комп'ютерам вчитися з даних, які не були явно запрограмовані [3].

Головною метою системи є узагальнення їх досвіду. Навчальний набір даних набуває досвіду для подальшого аналізу даних з більшою точністю.

Задачі машинного навчання, як правило, поділяють на дві категорії [14]:

— Навчання з вчителем: Комп'ютер представляє приклади входів і бажаних результатів, встановлених "вчителем", і метою є навчити загальне правило, яке відображає вхідні дані до результатів. У деяких випадках вхідний сигнал може бути частково доступним або може бути обмежений спеціальним зворотним зв'язком;

— Навчання без вчителя: Алгоритмічне навчання не дається етикеткам, залишаючи їх знайти структуру на вході. Навчання без вчителя може бути

самоціллю (виявлення прихованих у даних моделей) або засобу досягнення мети (атрибути навчання).

В свою чергу навчання з вчителем поділяють на:

— Автоматичний семінар: Комп'ютер видає неповний тренувальний сигнал: навчальний набір, який не має декількох цільових виходів (часто численні);

— Активне навчання: Комп'ютер може отримувати тільки навчальні мітки для обмеженого набору випадків (залежні від бюджету) і повинні оптимізувати вибір об'єктів для отримання міток. Для інтерактивного використання вони можуть бути надані користувачеві для тегу;

— Навчання з підкріпленням: Дані навчання (у вигляді винагород і покарань) надаються лише як реакція на програмні заходи в динамічній обстановці, наприклад, водіння автомобіля або відтворення супротивника.

Машинне навчання і видобування даних часто використовують ті ж самі методи, і багато їх перекриваються, але в той час як машинне навчання фокусується на прогнозуванні, заснованому на відомих властивостях, що вивчаються за допомогою навчальних даних, інтелектуальний аналіз даних, орієнтований на виявлення, ще невідомих властивостей (на етапі аналізу відкритих знань у базах даних)[16].

Інтелектуальний аналіз даних використовує багато методів машинного навчання, але для інших цілей; З іншого боку, він використовує методи машинного навчання та вилучення даних як "навчання не вчителя", або як етап попередньої обробки для підвищення точності механізму навчання.

В області автоматизованого навчання аналізу даних використовується метод, що використовується для прогнозування моделей і алгоритмів, які обслуговують комплекс прогнозування - комерційне застосування відоме як прогнозний аналіз. Ці аналітичні моделі дозволяють дослідникам, дослідникам даних, інженерам та аналітикам «виробляти надійні, повторювані та орієнтовані на результат рішення» і розкривати «приховане розуміння», вивчаючи історичні кореляції та тенденції в даних.

Кластеризація є одним із підходів машинного навчання.

3.2 Кластерний аналіз

«Кластеризація — це автоматичне розбиття елементів деякої множини на групи в залежності від їх схожості. Елементами безлічі можуть бути будь-які об'єкти, наприклад, дані або вектора характеристик. Самі ж групи прийнято також називати кластерами » [6].

Кластеризації часто порівнюють зі схожою процедурою класифікацією. Їх відмінність полягає в тому, що в класифікації безліч результуючих груп задано чітко, а в кластеризації воно визначається самим алгоритмом в процесі його роботи.

Існує безліч практичних застосувань кластеризації як в інформатиці, так і в інших областях. Ось кілька прикладів застосування кластеризації [7]:

— аналіз даних, спрощення роботи з інформацією, візуалізація даних;

— витяг і пошук інформації;

— угруповання і розпізнавання об'єктів.

Кластерний аналіз можна представити у вигляді такої послідовності дій [8]:

— вибір безлічі об'єктів;

— визначення безлічі змінних, для оцінки об'єктів і складання векторів характеристик;

— нормування векторів характеристик одним з доступних методів;

— визначити подібність між об'єктами по заданій метриці;

— застосування обраного методу кластерного аналізу для розбиття множини об'єктів на кластери по зі ступеня схожості;

— представлення результатів аналізу.

Проаналізувавши результати кластеризації можна скорегувати вибрані параметри, метрику або метод кластеризації, для поліпшення результатів. Проводячи дані поліпшення можна прийти до найкращого результату.

Основною метою кластерного аналізу є пошук груп подібних об'єктів у вибірці. Спектр застосування кластерного аналізу дуже широкий: він використовується в археології, антропології, медицині, психології, хімії, біології, державному управлінні, філології, маркетингу, соціології та інших дисциплінах. Однак універсальність застосування призвела до виникнення великої кількості несумісних термінів, методів і підходів, що ускладнюють однозначне використання і послідовне тлумачення кластерного аналізу.

Кластерний аналіз виконує наступні основні завдання:

— розробка типології або класифікації;

— дослідження корисних концептуальних схем групування об'єктів;

— породження гіпотез на основі дослідження даних;

— перевірка гіпотез або дослідження для визначення, чи дійсно групи, виділені тим чи іншим способом, присутні в наявних даних.

3.3 Класифікація алгоритмів кластеризвції

Плоскі алгоритми розбивають заданий безліч об'єктів на кластери, будуючи єдине розбиття, і для отримання іншого розбиття, необхідно повторювати процес кластеризації з іншими параметрами.

Ієрархічні алгоритми, на відміну від плоских, створюють не єдине розбиття, а систему вкладених розбиття на непересічні кластери. В результаті виконання цього алгоритму виходить дерево розбиття, коренем якого є кластер, що містить всі безліч об'єктів, а листям більш дрібні кластера.

Чіткі або нечіткі така класифікації визначають, чи може один об'єкт вибірки належати одночасно кільком кластерам, або він завжди належить єдиному кластеру [9].

У чітких (або непересічних) алгоритмах кожен об'єкт вибірки належить тільки одному кластеру, тобто кожному об'єкту зіставляється єдиний номер кластера, якому він належить.

В нечітких (або пересічні) алгоритмах кожному об'єкту зіставляється набір речових значень, що відображають ймовірність відносини даного об'єкта до кожного з кластерів. Іншими словами, в нечітких алгоритмах, кожен об'єкт належить всім кластерам з різним ступенем.

3.4 Об'єднання кластерів

Деякі алгоритми кластеризації виконують розбиття поступово, а саме на кожному кроці два найближче розташованих об'єкта об'єднуються і розглядаються як один кластер. У зв'язку з цим виникає необхідність запровадження заходів відстаней між кластерами, для визначення їх близькості[10].

Ось кілька способів, обчислення відстані між кластерами:

— одиночна зв'язок (Найближчий сусід). В даному способі, відстанню між кластерами вважається відстань між двома найбільш близькими об'єктами (найближчими сусідами) в різних кластерах. Тобто обчислюється відстані між усіма можливими парами об'єктів з різних кластерів і після цього обчислюється мінімальне з цих відстаней. Цей «мінімум» і вважається відстанню між кластерами;

— повна зв'язок (Найбільш віддалений сусід). Відстані між кластерами обчислюється аналогічно Одиночної зв'язку, але замість мінімальної відстані обчислюється максимальне, тобто відстань між найбільш віддаленими сусідами. Даний спосіб, як правило, працює досить добре, якщо кластери мають форму близьку до сферичної. Якщо ж кластери є «ланцюжковими» або мають подовжену форму, то цей метод непридатний;

— невиважені попарне середнє. Відстань між двома відмінними один від одного кластерами можна визначити, як середня відстань між усіма парами об'єктів з різних кластерів. Даний метод досить ефективний, за умови, що об'єкти формують різні групи, проте він працює також добре і в випадках, протяжних або «цепочечного» типу кластерів;

— виважена попарне середнє. Цей метод схожий з методом невиваженого попарного середнього, проте в ньому, при обчисленні відстаней враховується розмір відповідних кластерів, (тобто число об'єктів, що містяться в них), і використовується в якості вагового коефіцієнта. У зв'язку з цим даний метод необхідно використовувати, якщо очікуються кластери, значно відрізняються за розмірами;

— невиважені центроїдного метод. Для обчислення відстані цим методом, необхідно обчислити центри тяжкості кожного з кластерів, а після цього вважати за відстань між кластерами відстань між їх центрами тяжкості. При обчисленні центру ваги вага кожного об'єкта вважається рівним одиниці;

— зважений центроїдного метод (медіана). Цей метод ідентичний попередньому, за винятком того, що при обчисленнях використовуються ваги для обліку різниці між розмірами кластерів. Тому, якщо є або підозрюються значні відмінності в розмірах кластерів, цей метод виявляється переважно попереднього.

3.5 Алгоритми кластеризації

3.5.1 Алгоритми ієрархічної кластеризації

Алгоритми ієрархічної кластеризації прийнято розділяти на два типи: низхідні і висхідні алгоритми [5]. Спадні, діють за принципом «від більшого до меншого»: на початку процесу всі об'єкти поміщаються в єдиний кластер,

вершину дерева, після чого, на кожному кроці, один з існуючих кластерів розбивається на два дрібніших, поки кожен об'єкт не буде належати власним кластеру.

Другий тип алгоритмів, висхідні, більш поширений, і працює в зворотну сторону, щодо першого. Спочатку кожен з об'єктів поміщається в власний кластер, і на кожному кроці алгоритму, два найближчих кластера об'єднуються в один, до тих пір, поки не залишиться єдиний кластер, який містить всю вибірку об'єктів.

Результатом кластеризації даними алгоритмом є дерево розбиття, зване Дендрограмма. Найбільш популярний приклад використання ієрархічної кластеризації - класифікація тварин і рослин

3.5.2 Алгоритми квадратичної помилки

Завдання кластеризації можна інтерпретувати інакше: необхідно побудувати оптимальне розбиття об'єктів на групи.

Алгоритми даної категорії відносяться до класу плоских алгоритмів. Метод k-середніх вважається найбільш популярним в цій категорії, з огляду на те, що алгоритм розбиває задану безліч об'єктів на вказане число кластерів, розташованих на якомога більшій відстані один від одного. Робота цього методу розбивається на кілька етапів:

— випадково вибрати k початкових «центрів мас» кластерів;

— віднести кожен об'єкт до кластеру з найближчим «центром мас»;

— перерахувати «центримас» кластерів відповідно до їх поточним складом;

— перевірити критерій зупинки, і в разі його невиконання, повернутися до пункту 2.

Як критерій зупинки роботи алгоритму як правило використовують мінімальна зміна середньоквадратичної помилки. Також робота алгоритму завершується, якщо на кроці 2 не було об'єктів, які змінили свій кластер.

3.5.3 Нечіткі алгоритми

Як було сказано вище, алгоритми нечіткої кластеризації відносять кожен об'єкт до кожного кластеру з певною ймовірністю, на відміну від чітких методів. Алгоритмів даної категорії не дуже багато, з огляду на це розглянемо найбільш популярний метод c-середніх (c-means). Етапи роботи алгоритму схожі з етапами методу k-середніх:

— задати початкове нечітке розбиття n об'єктів на k кластерів шляхом вибору матриці приналежності U розміру n * k;

— знайти значення критерію нечіткої помилки, використовуючи матрицю U,

$$E^2(X,U) = \sum_{i=1}^{N}\sum_{j=1}^{K} U_{ij}\left|x_i^j - c_j\right|^2, c_k = \sum_{i=1}^{N} U_k x_i \qquad (3.1)$$

де X безліч об'єктів,x^j, x_i^j їх координати, c_j «центр мас» кластера j (вважаючи масу кожної точки дорівнює одиниці), U_{ij}– матриця приналежності;

— перегрупувати об'єкти з метою зменшення цього значення критерію нечіткої помилки;

— повертатися в п. 2 до тих пір, поки зміни матриці U не стануть незначними.

Цей алгоритм варто застосовувати тільки якщо заздалегідь відомо число кластерів і необхідно обчислити відношення кожного об'єкта до всіх кластерам

3.5.4 Алгоритм виділення зв'язкових компонент

Для роботи даного алгоритму необхідний параметр R, що задає граничне значення для ваг ребер. В процесі роботи цього алгоритму поступово видаляються всі ребра, вага яких перевищує порогове значення. В результаті роботи виходить граф, в якому залишаються тільки ребра, що з'єднують найбільш близькі об'єкти. 1

Щоб отримати кластери, залишається тільки підібрати значення R так, щоб граф розділився на кілька зв'язкових компонент, які і будуть кластерами.

Найчастіше, для підбору параметра R користуються побудовою гістограми розподілів попарних відстаней. Якщо кластерна структура виражена відносно добре, то гістограма буде мати два піки, один з яких відповідає внутрі кластерним відстаням, а другий меж кластерним. Параметр R підбирається із зони мінімуму між цими піками.

Мінус цього алгоритму в досить складному управлінні результуючим кількістю кластерів

4 АЛГОРИТМ АНАЛІЗУ ІНФОРМАЦІЇ ЩОДО ВИЗНАЧЕННЯ ПОТЕНЦІЙНО ПРОГРЕСИВНИХ ГРАВЦІВ

При розробці програмного продукту важливим чинником є розуміння математичної частини. Для вирішення задачі аналізу інформації щодо визначення потенційно прогресивних гравців.

4.1 Алгоритм кластеризації К-середніх

Класифікація середнього кластера (група k-середніх) є популярним методом групування, упорядкування множини об'єктів у відносно однорідних групах. Винайдено в 1950-х роках математиком Уго Штейнгаузом і майже одночасно Стюарт Ллойдом. Особливу популярність після звільнення Маккуїна.

Метою методу є розділення на кластеризаційні спостереження k, тому кожне спостереження належить до кластеру з найближчим середнім значенням. Метод заснований на мінімізації кількості квадратів відстаней між кожним спостереженням і його кластерним центром, що діє при поширенні математично акустичної хвилі математично описаних граничних задач для рівняння Гельмгольца зі складним несамоспряженим оператором[15].

$$\sum_{i=1}^{N} d(x_i, m_j(x_i))^2, \tag{4.1}$$

де d — метрика,x_i— і-ий об'єкт даних, $m_j(x_i)$а — центр кластера, якому на j-ій ітерації приписаний елемент x_i.

Ми маємо ряд спостережень (об'єктів), кожен з яких має певну цінність з ряду причин. Згідно з цими значеннями об'єкт розташований у багатовимірному просторі.

—дослідник визначає кількість створених груп;

—випадково обрані спостереження, які на даному етапі вважаються кластерними центрами;

—кожне спостереження «приписується» одному з російських кластерів - тобто найкоротшій відстані;

—розрахувати новий центр як частину кожної групи, знаки обчислюються як середнє арифметичне атрибутів об'єктів, включених до цієї групи;

—існує так багато ітерацій (повторіть кроки 3-4), поки центри кластера не стануть стабільними (наприклад, кожна ітерація в кожному класі опинятимуться тими ж об'єктами) в кластерних варіаціях мінімізована, а між кластерами – максимально.

Виберіть кількість кластерів на основі гіпотези дослідження. Якщо ні, то рекомендується створити дві групи 3,4,5, потім порівняти результати.

Приклад алгоритму:

Перший крок показаний на малюнку 4.1. k початковий "середній" (тут k = 3).

Другий крок полягає у створенні k ядер, що пов'язують кожне спостереження з найближчим оточенням. Розділ узгоджується з діалектною діаграмою Вороного. Рисунок 4.2.

На третьому етапі центроїд кожної з k груп стає новим середовищем. Третій крок показаний на рисунку 4.3.

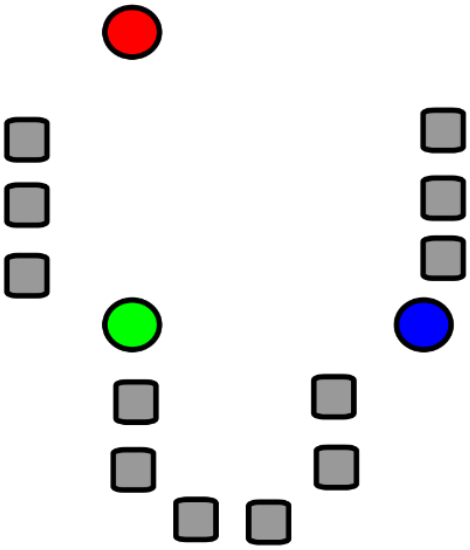


Рисунок 4.1 — Крок перший

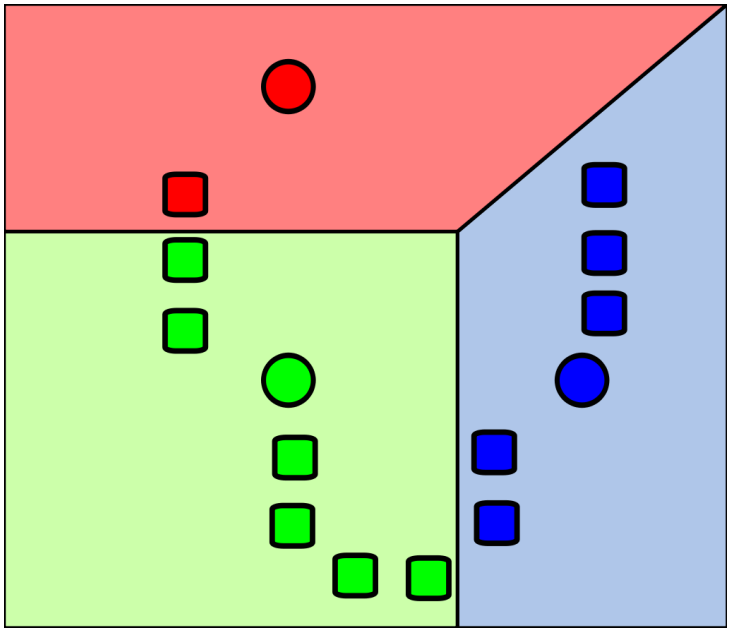


Рисунок 4.2 — Крок другий

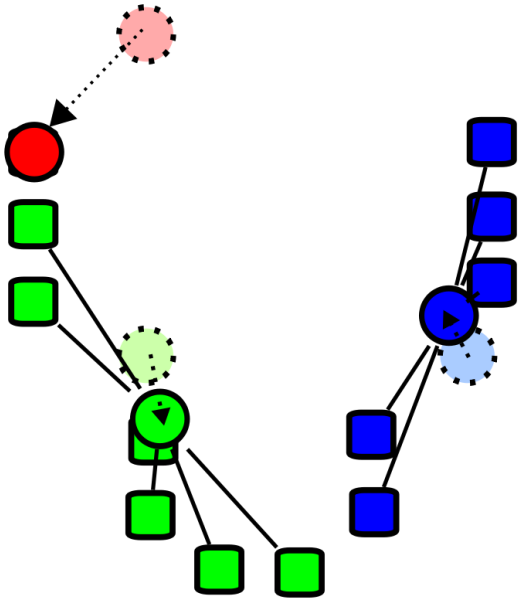


Рисунок 4.3 — Крок третій

На четвертому кроці ітерації 2 і 3 повторюються до досягнення збіжності. Крок четвертий зображено на рисунку 4.4.

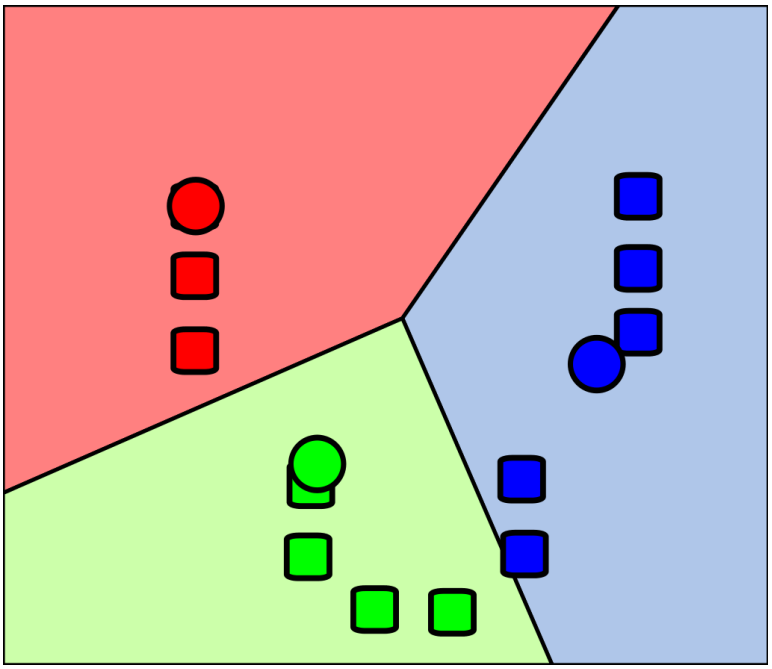


Рисунок 4.4 — Крок четвертий

Принцип роботи алгоритму полягає в пошуку таких кластерних центрів і наборів елементів кожного кластера за наявності функції F (°)[17], що виражає якість поточного розбиття множини на кластерах k, коли загальне квадратичне відхилення елементів кластера в центрах цих кластери будуть найменшими:

$$V = \sum_{i=1}^{k} \sum_{x_j \epsilon S_i} (x_j - \mu_i)^2 \qquad (4.2)$$

де k — число кластерів, S_i — отримані кластери, i = 1 , 2 , … , k, μ_i— центри мас векторів $x_j \in S_i$.

На початку алгоритму центри кластера вибираються випадковим чином, після чого для кожного елемента множини ітеративно обчислюють відстань від центрів шляхом складання кожного елемента в кластері з найближчим центром. Для кожної з отриманих груп розраховуються нові значення центрів, намагаючись мінімізувати функцію F (°), після чого повторюється процедура перерозподілу елементів між кластерами.

Алгоритм схеми "k-групування":

— вибір інформаційних точок k як центрів кластеризації до завершення процесу зміни кластерних центрів;

— для узгодження кожної інформаційної точки з групою, відстань до якої мінімальний;

— переконайтеся, що кожен кластер містить принаймні одну точку. Для цього кожен пустий кластер повинен бути заповнений довільною точкою, розташованою далеко від центру кластера;

— центр кожної групи замінюється середнім елементом кластера;

— кінець.

4.2 Переваги та недоліки методу k-середніх

Основні переваги k-середніх — простота і швидкість виконання. Метод групування k-means є більш зручним для великої кількості спостережень, ніж метод ієрархічного кластерного аналізу [5].

Одним з недоліків методу є просто розрив членів кластерних елементів зв'язку, так що розвиток різних змін методу та його нечітких аналогів, де перший етап алгоритму може бути елементом, що належить до декількох наборів кластерів (з різними ступенями належності).

Незважаючи на очевидні переваги способу, він має істотні недоліки [1]:

— результат класифікації значною мірою залежить від випадкового початкового положення центрів розгону;

— алгоритм чутливий до викидів, які можуть спотворювати носії;

— кількість кластерів повинна бути визначена дослідником.

5 Програмні рішення для реалізації алгоритму K-means

5.1 Додатки в Matlab

FuzzyLogicToolbox надає функції, програми та блок Simulink для аналізу, проектування та моделювання систем на основі нечіткої логіки. Продукт проведе вас через етапи проектування систем нечіткого виводу. Функції передбачені для багатьох загальних методів, включаючи нечітке кластеризацію.

Панель інструментів дозволяє моделювати складні поведінки системи, використовуючи прості логічні правила, а потім реалізувати ці правила в системі нечіткого виводу. Ви можете використовувати його як автономний нечіткий механізм виводу. Альтернативно, можна використовувати блоки нечіткого виводу в Simulink і імітувати нечіткі системи в рамках комплексної моделі всієї динамічної системи.

5.2 Python бібліотеки scikit-learn та SciPy

SciPy безкоштовна бібліотека Python з відкритим вихідним кодом, яка використовується для наукових обчислень і технічних обчислень[13].

SciPy містить модулі для оптимізації, лінійної алгебри, інтеграції, інтерполяції, спеціальних функцій, FFT, обробки сигналів і зображень, вирішувачів ОДУ та інших загальних для науки і техніки завдань.

SciPy будує на об'єкті масиву NumPy і є частиною стека NumPy, що включає в себе такі інструменти, як Matplotlib, pandas і SymPy, а також розширений набір бібліотек для наукових обчислень. Цей стек NumPy має подібних користувачів до інших програм, таких як MATLAB, GNU Octave і Scilab. Стек NumPy також іноді називається стеком SciPy.

Бібліотека SciPy в даний час поширюється під ліцензією BSD, і її розробка спонсорується і підтримується відкритою спільнотою розробників. Він також підтримується NumFOCUS, фундацією громади для підтримки відтворюваної та доступної науки.

Scikit-learn бібліотека для вільного програмного навчання для мови програмування Python [18]. Він має різні алгоритми класифікації, регресії та кластеризації, включаючи векторні машини підтримки, випадкові ліси, градієнтні підсилення, k-середні та DBSCAN, а також призначений для взаємодії з чисельними та науковими бібліотеками Python та NumPy [5].

Scikit-learn це багате середовище для забезпечення сучасних реалізацій багатьох відомих алгоритмів машинного навчання, зберігаючи при цьому простий у використанні інтерфейс. Це відповідає зростаючій потребі у статистичному аналізі даних неспеціалістами в галузі програмного забезпечення та веб-індустрії, а також у галузях поза комп'ютерними науками, такими як біологія або фізика.

5.3 Мова програмування R

R - мова програмування та середовище вільного програмного забезпечення для статистичних обчислень і графіки, що підтримуються Фондом R для статистичних обчислень.

Мова R широко використовується серед статистиків і шахтарів даних для розробки статистичного програмного забезпечення і аналізу даних. Опитування, опитування інтелектуального аналізу даних та дослідження баз даних наукової літератури свідчать про значне зростання популярності за останні роки. Попередньо скомпільовані двійкові версії надаються для різних операційних систем. Хоча R має інтерфейс командного рядка, існує декілька графічних

інтерфейсів користувача, наприклад, RStudio, інтегрованого середовища розробки.

R був створений Россом Іхакою та Робертом Джентльменом в Університеті Окленду, Нова Зеландія, і в даний час розробляється основною командою R Development (члени якої є членами). R названий частково після перших імен перших двох авторів R. Проект був задуманий у 1992 році, з початковою версією, випущеною у 1995 році, і стабільною бета-версією у 2000 році.

R та його бібліотеки реалізують широкий спектр статистичних та графічних методів, включаючи лінійне та нелінійне моделювання, класичні статистичні тести, аналіз часових рядів, класифікацію, кластеризацію та інші. R легко розширюється за допомогою функцій і розширень, і спільнота R відзначається своїми активними вкладами в термінах пакетів. Багато стандартних функцій R написані на самому R, що дозволяє користувачам легко дотримуватися алгоритмічних рішень.

Для інтенсивних обчислювальних завдань код C, C ++ і Fortran можна зв'язати і викликати під час виконання. Просунуті користувачі можуть писати C, C ++, Java, .NET або Python-код, щоб маніпулювати об'єктами R безпосередньо. , Можливості імпорту / експорту, інструменти звітності (knitr, Sweave) і т.д. Ці пакети розроблені в основному на R, а іноді на Java, C, C ++ і Fortran.

5.4 Accord.NET

Accord.NET - це основа для наукових обчислень у .NET. Вихідний код проекту доступний на умовах ліцензії GnuLesserPublicLicense версії 2.1.

Структура містить набір бібліотек, які доступні у вихідному коді, а також за допомогою виконуваних інсталяторів та пакетів NuGet. Основними областями, що охоплюються, є чисельна лінійна алгебра, чисельна оптимізація, статистика, машинне навчання, штучні нейронні мережі, обробка сигналів і зображень, бібліотеки підтримки (такі як графік і візуалізація). Спочатку проект

був розроблений для розширення можливостей платформи AForge.NET Framework, але з того часу в неї ввійшов AForge.NET. Нові версії об'єднали обидві фреймворки під назвою Accord.NET.

Корпорація Accord.NET Framework була представлена в декількох книгах, таких як Mastering.NET Машинне навчання PACKT публікації і F # для машинного навчання програм, показані в QCON Сан-Франциско.

6 ЗАСОБИ РОЗРОБКИ

При розробці програмного продукту важливим чинником є правильний вибір технологій та засобів програмної реалізації. Основним середовищем розробки було Spyder.Для створення інтерфейсу використовувалися QtDesigner. Для розробки алгоритмів використовувалась мова Python на базі бібліотек sklearn.

Для робот з файлами була використана бібліотека pandas для Python[12].

6.1 Середовище розробки Spyder

Spyder (колишній Pydee) - вільна і інтерактивна кросплатформена IDE (Integrated Development Environment) для наукових обчислень в Python, що забезпечує простоту використання і легкість програмного забезпечення.

Spyder є частиною Python модуля spyderlib, заснованого на PyQt4, pyflakes, rope і Sphinx, який забезпечує потужні віджети PyQt4, такі як редактор коду, консоль Python (вбудований додаток), графіка редактора змінних (включаючи списки, словники і матриці).

Особливості:

—Python, C / C ++ і редактор підсвічування синтаксису Fortran;

—динамічний інтроспекційний код (за допомогою канату) - завершення переходу для визначення об'єкта при натисканні миші;

—виявлення помилок під час польоту (за допомогою затяжок);

—підтримує використання багатьох консолей Python (включаючи оболонку IPython);

—доступ до документації (у форматі Sphinx);

—інтеграція з академічними бібліотеками Python - NumPy, SciPy, Matplotlib, Pandas;

—переглядати та редагувати змінні за допомогою графічного інтерфейсу (підтримує різні типи даних - числа, рядки, списки, інформаційні панелі, словники).

Творці продовжують дбати про поліпшення продуктивності розробників, створюючи всі умови, щоб зосередитися тільки на написанні коду. Наприклад, вдосконалення вже захоплених функцій перегляду коду, виправлення та коригування коду.

6.2 QtDesigner

QtDesigner - без платформ середовище для розробки графічного інтерфейсу користувача (GUI) з використанням бібліотеки Qt. Входить до складу Qt.

Дизайнер Qt дозволяє створювати графічні інтерфейси користувача з рядом інструментів. Є панель інструментів віджетів, яка дозволяє використовувати елементи інтерфейсу - віджети, такі як ComboBox, FieldEdit Field, PushButton і багато іншого. Кожен віджет має свій власний набір властивостей, який визначається відповідним классом бібліотеки Qt. Властивості віджетів можна змінювати за допомогою редактора властивостей. Для кожного классу віджетів існує спеціалізований видавець. Характерною особливістю конструктора Qt є підтримка візуального редагування сигналів і слотів. Наприклад, можна зв'язати сигнал, який генерується, перемикаючи стан віджета CheckBox у слот для наявності іншого віджета.

Дизайнер Qt може бути запущений як окрема програма, а також як вбудований IDE (Integrated Development Environment) QtCreator.

6.3PyQt5

PyQt - це набір "зв'язків" графічної рамки Qt для мови програмування Python, яка виконується як розширення Python.

PyQt працює на всіх платформах, що підтримуються Qt: Linux та іншими операційними системами, такими як UNIX, MacOSX і Windows.

PyQt також включає QtDesigner, дизайнера графічного інтерфейсу. Програма pyui сгенерує код Python з файлів, створених у DesignerQt. Це робить PyQt дуже корисним інструментом для швидкого прототипування. Ви також можете додати нові графічні елементи керування, написані на Python для дизайнера Qt.

6.4Python

Python є інтерпретованою мовою програмування високого рівня загального призначення. Створений Гвідо ван Россумом і вперше випущений в 1991 році, філософія дизайну Python підкреслює читабельність коду з його помітним використанням значних пробілів. Його мовні конструкції та об'єктно-орієнтований підхід спрямовані на те, щоб допомогти програмістам написати чіткий, логічний код для малих і великих проектів.

Python динамічно набирається і збирається сміттям. Він підтримує багато парадигм програмування, включаючи процедурні, об'єктно-орієнтовані та функціональні програми. Python часто описується як "батарея включена" мова через його всеосяжну стандартну бібліотеку.

Інтерпретатори Python доступні для багатьох операційних систем. Світове співтовариство програмістів розробляє та підтримує CPython. Некомерційна організація, Python Software Foundation, управляє та направляє ресурси для розвитку Python та CPython [2].

Python може служити мовою сценаріїв для веб-додатків, наприклад, через mod_wsgi для веб-сервера Apache. За допомогою інтерфейсу Web Server Gateway розроблено стандартний API(Application Programming Interface) для полегшення цих програм. Веб-фреймворки як Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle і Zope підтримують розробників у розробці та обслуговуванні складних додатків. Pyjs і IronPython можна використовувати для розробки клієнтських додатків на основі Ajax. SQLAlchemy може використовуватися як перетворювач даних у реляційну базу даних. Twisted - це програма для комунікації між комп'ютерами і використовується (наприклад) Dropbox.

Бібліотеки, такі як NumPy, SciPy і Matplotlib, дозволяють ефективно використовувати Python в наукових обчислень, із спеціалізованими бібліотеками, такими як Biopython і Astropy, що забезпечують функціональність, специфічну для домену. SageMath - це математичне програмне забезпечення з інтерфейсом для ноутбуків, програмується в Python: його бібліотека охоплює багато аспектів математики, включаючи алгебру, комбінаторику, чисельну математику, теорію чисел і числення.

7 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

7.1 Інсталяція та системні вимоги

Для встановлення розробленої програмної системи персональний комп’ютер повинен мати процесор Intel ® Core ™ 2 / 2 Quad / Pentium ® / Celeron ® / Xeon™ чиAMD 6 / Turion ™ / Athlon ™ / Duron ™ / Sempron ™з тактовою частотою не нижче2 GHz, на комп’ютері повинна бути встановлена операційна система Windows 7,Windows 8, Windows 10. Також на жосткому диску повинно бути вільне місце.

7.2Інструкція з використання програмного продукту

При запуску програми з’являється форма. На рисунку 7.1 зображена головна форма.

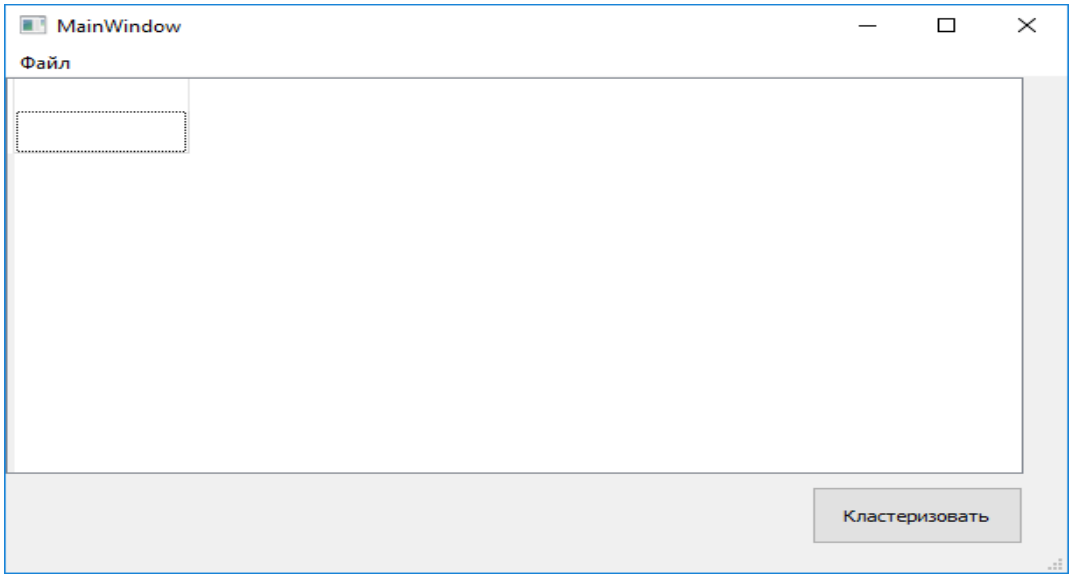


Рисунок 7.1 — Головна форма

При натисканні на кнопку меню «Файл» буде відкрито випадаюче

меню.На рисунку 7.2 зображено випдаюче меню.

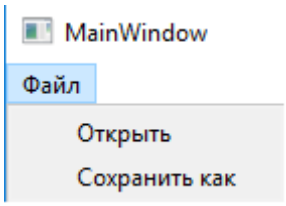


Рисунок 7.2 — Випадаюче меню

Якщо натиснути на кнопку «Открыть» тоді відкриється діалогове вікно для вибору та відкриття файлу. При натисканні кнопки «Сохранить как»також відкриться діалогове вікно але для збереження файлу у xlsx форматі. На рисунку 7.3 зображено діалогове вікно відкриття файлу.

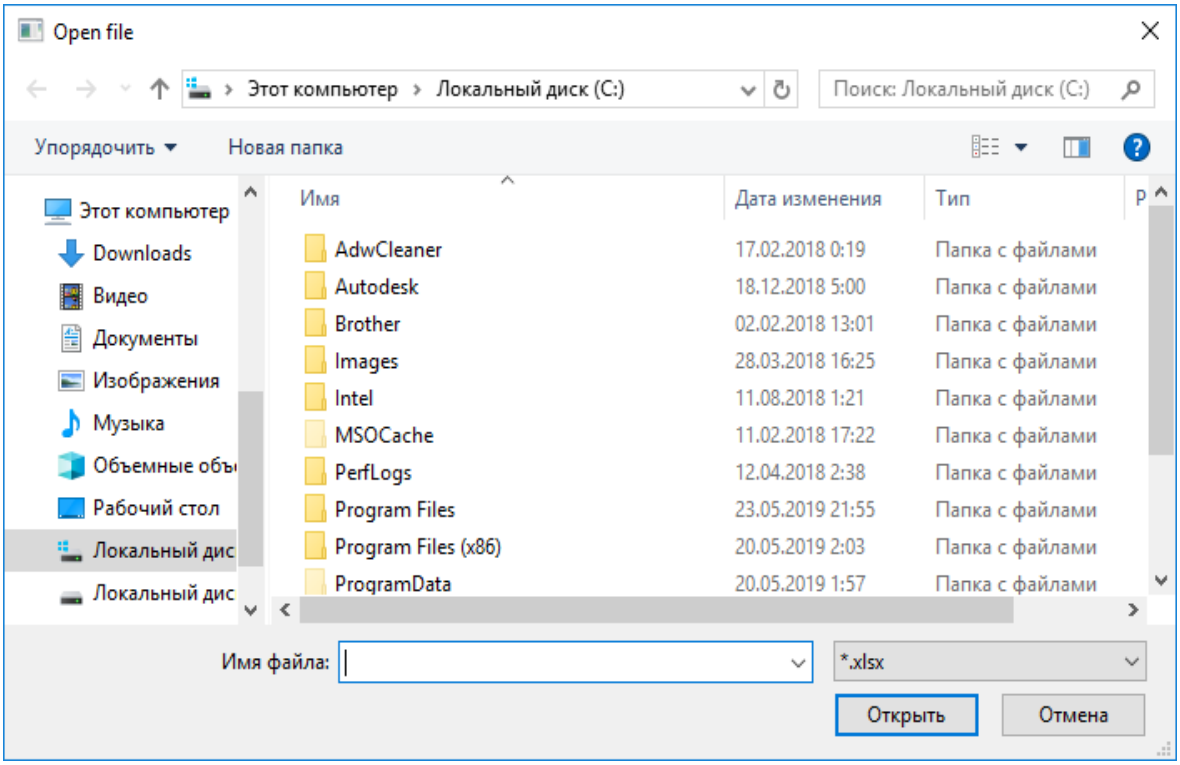
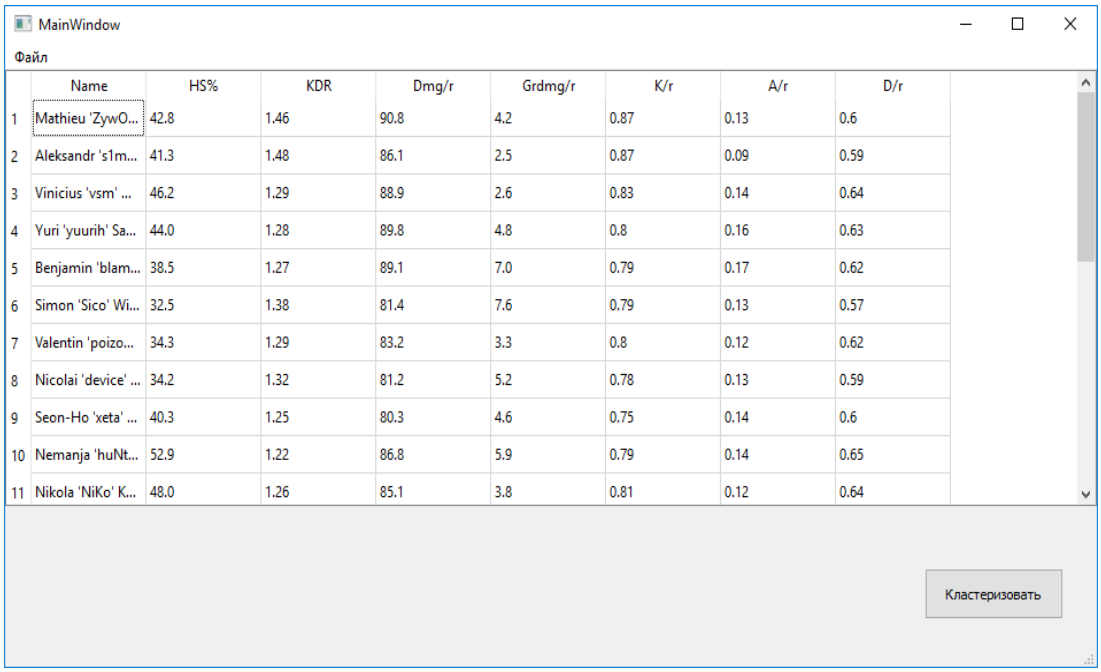


Рисунок 7.3 — Діалогове вікно відкриття файлу

Коли файл буле відкрито то головній формі можна побачити дані таблиці та при натисканні кнопки «Кластеризовать» відбудеться обробка даних таблиці

та додано нової інформації щодо розбитку на класи. На рисунку 7.4 зображено табицю вхідних даних для аналізу.



	Name	HS%	KDR	Dmg/r	Grdmg/r	K/r	A/r	D/r
1	Mathieu 'ZywO...	42.8	1.46	90.8	4.2	0.87	0.13	0.6
2	Aleksandr 's1m...	41.3	1.48	86.1	2.5	0.87	0.09	0.59
3	Vinicius 'vsm' ...	46.2	1.29	88.9	2.6	0.83	0.14	0.64
4	Yuri 'yuurih' Sa...	44.0	1.28	89.8	4.8	0.8	0.16	0.63
5	Benjamin 'blam...	38.5	1.27	89.1	7.0	0.79	0.17	0.62
6	Simon 'Sico' Wi...	32.5	1.38	81.4	7.6	0.79	0.13	0.57
7	Valentin 'poizo...	34.3	1.29	83.2	3.3	0.8	0.12	0.62
8	Nicolai 'device' ...	34.2	1.32	81.2	5.2	0.78	0.13	0.59
9	Seon-Ho 'xeta' ...	40.3	1.25	80.3	4.6	0.75	0.14	0.6
10	Nemanja 'huNt...	52.9	1.22	86.8	5.9	0.79	0.14	0.65
11	Nikola 'NiKo' K...	48.0	1.26	85.1	3.8	0.81	0.12	0.64

Рисунок 7.4 — Таблиця вхідних даних для аналізу

Після натискання кнопки «Кластеризовать» буде відкрите діалогове вікно, яке сповістить що кластеризація даних відбулася. На рисунку 7.5 зображено, що кластеризація виконана.

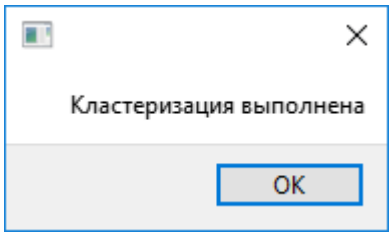
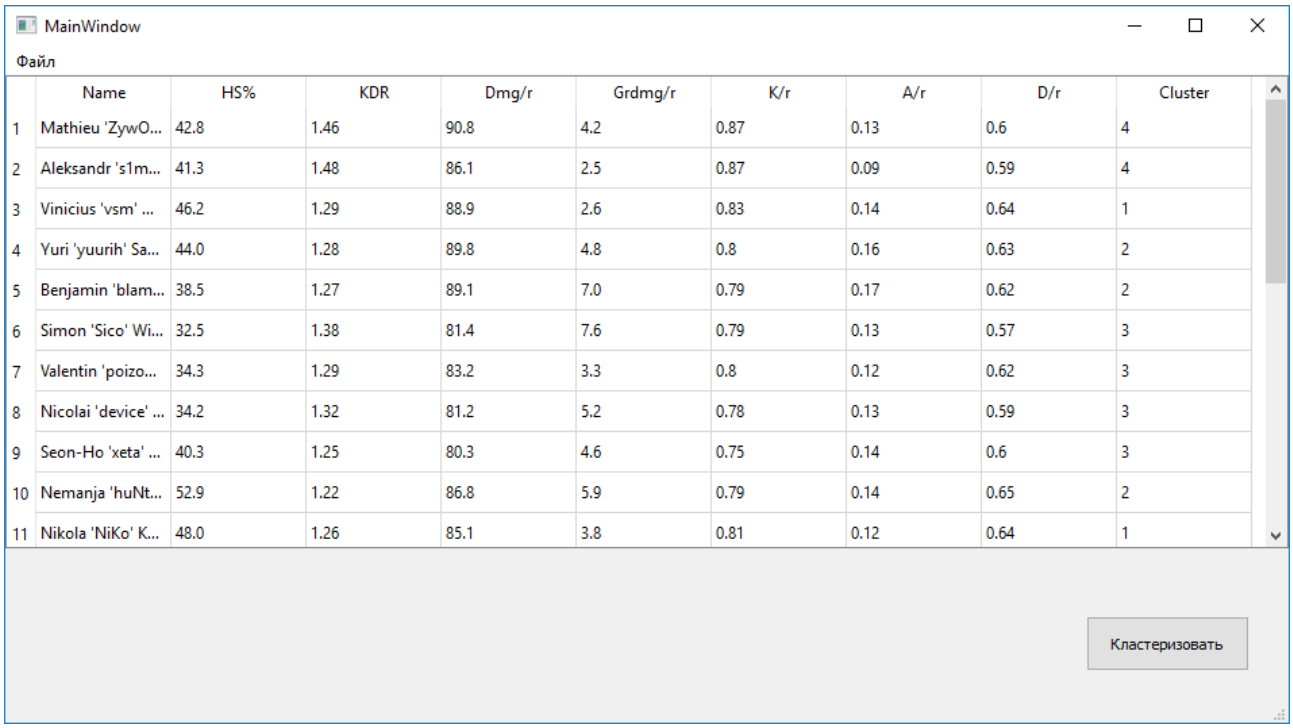


Рисунок 7.5 — Діалогове вікно, яке сповіщає, що кластеризація виконана

Після цього дані на головній формі будуть оновлені. На рисунку 7.6 зображено кластеризовані дані.



	Name	HS%	KDR	Dmg/r	Grdmg/r	K/r	A/r	D/r	Cluster
1	Mathieu 'ZywO...	42.8	1.46	90.8	4.2	0.87	0.13	0.6	4
2	Aleksandr 's1m...	41.3	1.48	86.1	2.5	0.87	0.09	0.59	4
3	Vinicius 'vsm' ...	46.2	1.29	88.9	2.6	0.83	0.14	0.64	1
4	Yuri 'yuurih' Sa...	44.0	1.28	89.8	4.8	0.8	0.16	0.63	2
5	Benjamin 'blam...	38.5	1.27	89.1	7.0	0.79	0.17	0.62	2
6	Simon 'Sico' Wi...	32.5	1.38	81.4	7.6	0.79	0.13	0.57	3
7	Valentin 'poizo...	34.3	1.29	83.2	3.3	0.8	0.12	0.62	3
8	Nicolai 'device' ...	34.2	1.32	81.2	5.2	0.78	0.13	0.59	3
9	Seon-Ho 'xeta' ...	40.3	1.25	80.3	4.6	0.75	0.14	0.6	3
10	Nemanja 'huNt...	52.9	1.22	86.8	5.9	0.79	0.14	0.65	2
11	Nikola 'NiKo' K...	48.0	1.26	85.1	3.8	0.81	0.12	0.64	1

Рисунок 7.6— Таблиця з кластеризованими даними

Тепер потрібно зберегти ці дані за допомогою кнопки «Сохранить как», яка знаходиться у випадаючому меню. Дані будуть збереженні у форматі xlsx. На рисунку 7.7 зображено діалогове вікно відкриття файлу.

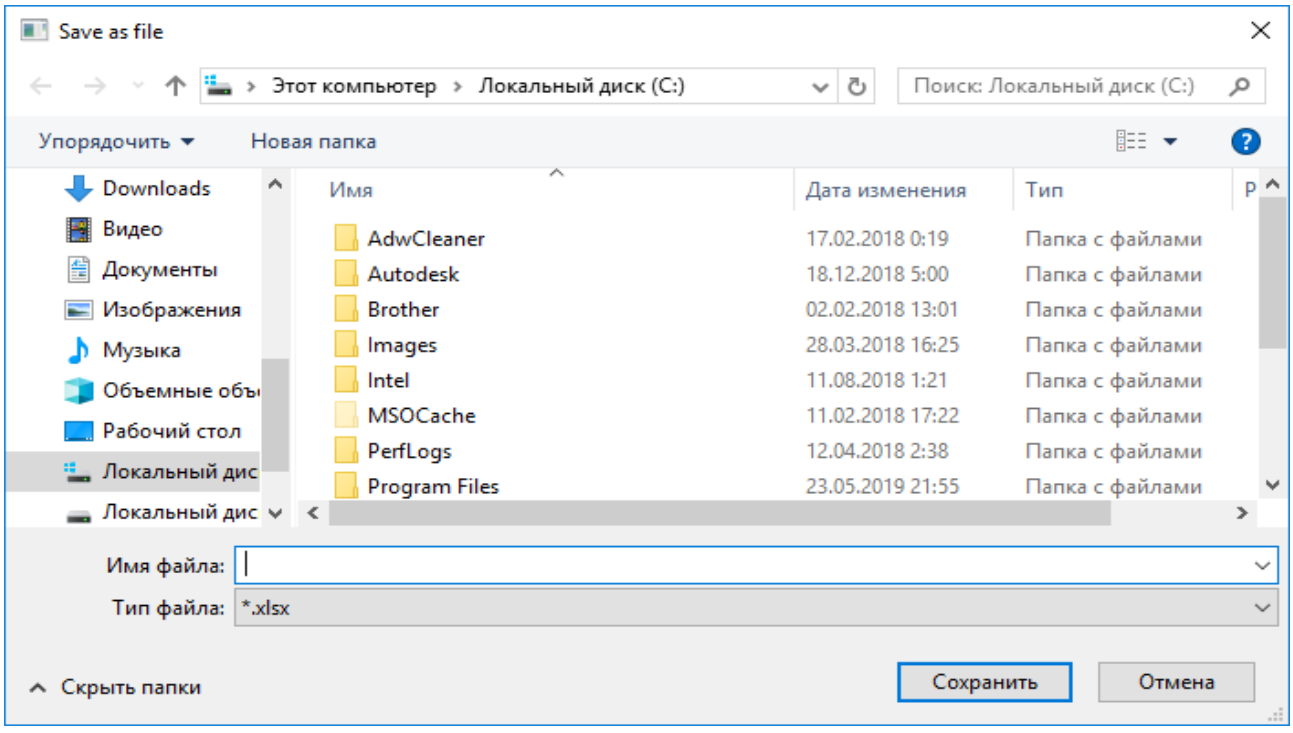


Рисунок 7.7 — Діалогове вікно збереження файлу

ВИСНОВКИ

У ході практики було поліпшено навички розробки програмних продуктів на мові Python, на платформах PyQt5. В ході виконання даної роботи було розроблено модуль для потенційно прогресивних гравців для рекомендації їх командам за допомогою метода k-means.

Модуль написано на мові Pythonта інтерфейс реалізований за допомогою QT Designer та PyQt5 набору бібліотек.

Модуль може бути використаний для більш масштабного проекту, який спеціалізується на визначенні потенційно прогресивних гравців.

Користувач надає вхідні дані з параметрами, які надалі аналізуються та обробляються. На виході генерується таблиця результатом, якої є оброблені вхідні дані, які розподілилися по класам та представлена у вигляді xlsx-файлу.

Користувач має змогу зберегти копію xslx-файлу, змінюючи його ім’я та вибравши директорію для збереження та може перевірити результат.

Дана система спрощує процесаналізу великої кількості даних гравців та допомагає розподілити їх на класи і визначити найбільш потенційно прогресивних гравців.

Практика покращила навички в програмуванні та надала досить вагомих знань в алгоритмах машинного навчання та обробки даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sayak Paul— K-Means Clusteringin Python with scikit-learn [Електронний ресурс]. — 2018. — Режим доступу: https://www.datacamp.com/community/tutorials/k-means-clustering-python.
2. Swaroop C H. A Byte of Python[Електронний ресурс]— 2013. — Режим доступу: http://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.01.pdf
3. Alex Smola and S.V.N. Vishwanathan — INTRODUCTION TO MACHINE LEARNING [Електронний ресурс]. — 2008. — Режим доступу: http://alex.smola.org/drafts/thebook.pdf.
4. Andrea Trevino — Introduction to K-means Clustering [Електронний ресурс]. — 2016. – Режим доступу: https://www.datascience.com/blog/k-means-clustering.
5. AurélienGéron — Hands-On Machine Learning with Scikit-Learnand Tensor Flow: Concepts, Tools, and Techniques to Build Intelligent Systems [Електронний ресурс]. — 2015. — Режим доступу: https://www.academia.edu/37010160/Hands-On_Machine_Learning_with_Scikit-Learn_and_TensorFlow.
6. A.K.Jain,M.N.Murty,P.J.Flynn—Data Clustering:A Review[Електронний ресурс]. —Режим доступу: http://www.csee.umbc.edu/nicholas/clustering/p264-jain.pdf.
7. Суслов, С. А. Кластерный анализ: сущность, преимущества и недостатки. — 2011. — С. 51-56.
8. Ершов, К. С. Анализ и классификация алгоритмов кластеризации. // Новые информационные технологии в автоматизированных системах. — 2016. — No19. — С. 274-279

9. Бериков, В. С. Современные тенденции в кластерном анализе / В. С. Бериков, Г. С. Лбов. — Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", —2008. —26 с.
10. Duran, B. S. Cluster Analysis—A Survey / B. S. Duran, P. L. Odell.Springer— 1974.
11. Мандель, И. Д. Кластерный анализ //Финансы и статика —1988. — 176 с.
12. Lutz, M. Programming Python //O'Reilly Media — 1996.
13. Blanco — Silva, F. J.LearningSciPy for Numerical and Scientific Computing . —Packt publishing, 2015.
14. Müller, A. C. Introduction to Machine Learning with Python: A Guide for Data Scientists.— O'Reilly Media — 2016.
15. Paul S. Bradley, Usama M. Frayyad — Refining Initial Points for K-Means Clustering [Електронний ресурс]. — Режим доступу: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.8528&rep=rep1&type=pdf.
16. Jasper Snoek, Hugo Larochelle, Ryan P. Adams — Practical Bayesian Optimization of MachineLearning Algorithms [Електронний ресурс]. — Режим доступу: http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.
17. Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schroledl — Constrained K-means Clustering with Background Knowledge [Електронний ресурс]. — Режим доступу: https://web.cse.msu.edu/~cse802/notes/ConstrainedKmeans.pdf.
18. Fabian Pedregosa, Ga¨el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel — Scikit-learn: Machine Learning in Python [Електронний ресурс]. — Режим доступу: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

ДОДАТОК А

Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення найбільш потенційно прогресивних гравців/команд та їх рекомендації для відповідних команд.

Специфікація

УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5154_19Б

Аркушів 2

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5154_19Б	Записка	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5154_19Б 12-1	Players.py	Модуль основного тіла програми
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5154_19Б 13-1	Players.py	Виконання алгоритму оцінки гравців

ДОДАТОК Б

Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення найбільш потенційно прогресивних гравців/команд та їх рекомендації для відповідних команд.

Текст програми

УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5154_19Б 12-1

Аркушів 3

Київ 2019

Текст програми

```
# -*- coding: utf-8 -*-
import pandas as pd
import xlrd
from sklearn.cluster import KMeans
from sklearn import preprocessing
import sys
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QFileDialog, QMessageBox


class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(950, 500)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(800, 400, 121, 41))
        self.pushButton.setObjectName("pushButton")
        self.tableWidget = QtWidgets.QTableWidget(self.centralwidget)
        self.tableWidget.setGeometry(QtCore.QRect(0, 0, 950, 350))
        self.tableWidget.setObjectName("tableWidget")
        self.tableWidget.setColumnCount(0)
        self.tableWidget.setRowCount(0)
        item = QtWidgets.QTableWidgetItem()
        self.tableWidget.setVerticalHeaderItem(0, item)
        item = QtWidgets.QTableWidgetItem()
        self.tableWidget.setHorizontalHeaderItem(0, item)
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 600, 21))
        self.menubar.setObjectName("menubar")
        self.menu = QtWidgets.QMenu(self.menubar)
        self.menu.setObjectName("menu")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)
        self.action = QtWidgets.QAction(MainWindow)
        self.action.setObjectName("action")
        self.action_2 = QtWidgets.QAction(MainWindow)
        self.action_2.setObjectName("action_2")
        self.action_3 = QtWidgets.QAction(MainWindow)
        self.action_3.setObjectName("action_3")
        self.menu.addAction(self.action)
        self.menu.addAction(self.action_3)
        self.menubar.addAction(self.menu.menuAction())

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.pushButton.setText(_translate("MainWindow", "Кластеризовать"))
        self.menu.setTitle(_translate("MainWindow", "Файл"))
        self.action.setText(_translate("MainWindow", "Открыть"))
        self.action_2.setText(_translate("MainWindow", "Сохранить"))
        self.action_3.setText(_translate("MainWindow", "Сохранить как"))

class MainWindow (QtWidgets.QMainWindow):
    def __init__(self, parent=None):
        QtWidgets.QWidget.__init__(self, parent)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

        self.ui.pushButton.clicked.connect(self.Cluster)
        self.ui.action.triggered.connect(self.Open)
        self.ui.action_3.triggered.connect(self.SaveAs)
        self.ui.tableWidget.setSortingEnabled(True)

    def Open(self):
        self.ui.tableWidget.setRowCount(0)
        global path
```

```
        path =  QFileDialog.getOpenFileName(self, 'Open file', '/','*.xlsx')[0]
        book = xlrd.open_workbook(path)
        sheet = book.sheets() [0]
        data = [[sheet.cell_value(r,c) for c in range (sheet.ncols)]for r in range(sheet.nrows)]
        header = data[0]
        #print(header)
        data=data[1:]
        #print(data)
        for row  , columnvalues in enumerate(data):
            self.ui.tableWidget.insertRow(row)
            self.ui.tableWidget.setColumnCount(len(columnvalues))
            for column , value in enumerate(columnvalues):
                item = QtWidgets.QTableWidgetItem(str(value))
                self.ui.tableWidget.setItem( row, column, QtWidgets.QTableWidgetItem(item))
        self.ui.tableWidget.setHorizontalHeaderLabels(header)
        return path

    def SaveAs(self):
        p =  QFileDialog.getSaveFileName(self, 'Save as file', '/','*.xlsx')[0]
        file = pd.ExcelWriter(p)
        dataK.to_excel(file)
        print(dataK)
        file.save()

    def Cluster(self):
        self.ui.tableWidget.setRowCount(0)
        data = pd.read_excel(path)
        datacl=data.drop(data.columns[0], axis=1).values
        dataNorm = preprocessing.scale(datacl)
        km = KMeans(n_clusters=4).fit(dataNorm)
        labels = km.labels_ +1
        global dataK
        dataK=data
        dataK['Cluster']=labels
        datal=dataK.get_values()
        header = dataK.axes
        h = header[1]
        for row  , columnvalues in enumerate(datal):
            self.ui.tableWidget.insertRow(row)
            self.ui.tableWidget.setColumnCount(len(columnvalues))
            for column , value in enumerate(columnvalues):
                item = QtWidgets.QTableWidgetItem(str(value))
                self.ui.tableWidget.setItem( row, column, QtWidgets.QTableWidgetItem(item))
        self.ui.tableWidget.setHorizontalHeaderLabels(h)
        msg = QMessageBox()
        msg.setWindowTitle(' ')
        msg.setText('Кластеризация выполнена')
        msg.exec()
        return dataK


if __name__=="__main__":
    app = QtWidgets.QApplication(sys.argv)
    myapp = MainWindow()
    myapp.show()
    sys.exit(app.exec_())using System;
```

ДОДАТОК В

Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення найбільш потенційно прогресивних гравців/команд та їх рекомендації для відповідних команд.

Опис програми

УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5154_19Б 13-1

Аркушів 8

Київ 2019

АНОТАЦІЯ

Метою роботи є створення системи для автоматизації процесу аналізу даних та розрахунків щодо визначення потенційно прогресивних гравців за вхідними параметрами.

Робота розробленої системи демонструється для отримання кінцевого файлу в якому міститься інформація про гравців розподілених на декілька класів.

Модуль написано на мові Pythonта інтерфейс реалізований за допомогою QT Designer та PyQt5 набору бібліотек.

Модуль може бути використаний для більш масштабного проекту, який спеціалізується на визначенні потенційно прогресивних гравців.

ЗМІСТ

ЗАГАЛЬНІ ВІДОМОСТІ

У цьому додатку міститься опис програмного продукту та його призначення. У додатку Б міститься програмний код.

Розроблена програма працює в операційній системі Windows 7, Windows 8 та Windows10 і потребує встановленого на ПК пакету програм Python.

При розробці програмного продукту використовувалась мова Python та середа розробки Spyder.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблений компонент призначений для аналізу інформації щодо визначення найбільш потенційно прогресивних гравців.

Поділяє гравців на класи, що дає змогу виявити потенційних гравців. Зменшує кількість часу аналізу статистки гравців, що дозволяє більш швидше і точніше оцінювати їх можливості.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Було розроблено програмний продукт, основною задачею якого є аналіз інформації щодо визначення найбільш прогресивних гравців.

Було показано та задіяно можливості даного програмного продукту, котрий полегшує аналіз інформації, а також поділяє гравців на групи, що надає змогу визначити потенційно прогресивних гравців.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для забезпечення повноцінної роботи та досягнення високої ефективності роботи створеної програми було обрано середу розробки Spyder на мові програмування Python, яка показала себе надійною та гнучкою середою розробки програм.

Розроблені додатки працюють в операційних системах Windows 7, Windows 8 та Windows10 і потребує встановленого на ПК пакету програм Python.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними є статистика гравців, яка записана у xlsx файлі.

Вихідними даними є проаналізована та оброблена інформація про статистику гравців і поділено їх на класи та ця інформація зберігається у xlsx файлі.